International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

Design OFMCM Methods for FIR Filter Architectures

SONGALA. SAILAJA VANI*,V. RAMA RAO
Songala.SalilajaVani,M-Tech(VLSI Design)ECE Department, Eluru College of Engineering And Technology

V.Rama Rao, M-Tech, MIAENG-Assistant Professor ECE Department, Eluru College of Engineering And Technology

Abstract—*Finite Impulse Response (FIR) filters are widely applied in multistandard wireless communications. A novel efficient algorithms and architectures have been introduced for the design of low complexity bit-parallel multiple constant multiplications (MCM) operation which dominates the complexity of many digital signal processing systems. In digit-serial MCM design that offers low complexity MCM operations that offers a low delay. In this previous design a MCM operations performed by CSE algorithm but it occupies large and delay area. In MCM design based Graph algorithmprovides low area and delay. In this we proposed a graph based multipliers i.e CSD, MSD,MAG based on graph architecture for implementing low complexity higher order FIR filters.*

Keywords: FIR-finite impulse response filters, CSD – Canonic Signed-Digit multiplier ,CSE- common subexpression elimination algorithms, MSD – Minimum Signed-Digit multiplier,MAG-Minimum added Graph Multiplier.

I. INTRODUCTION

FINITE impulse response (FIR) filters are of greatimportance in digital signal processing (DSP) systemssince their characteristics in linear-phase and feed-forwardimplementations make them very useful for building stablehigh-performance filters. The direct and transposed-form FIRfilter implementations are illustrated in Fig. 1(a) and (b),respectively. Although both architectures have similar complexityin hardware, the transposed form is generally preferredbecause of its higher performance and power efficiency [1].
The multiplier block of the digital FIR filter in its transposedform [Fig. 1(b)], where the multiplication of filter coefficientswith the filter input is realized, has significant impact onthe complexity and performance of the design because alarge number of constant multiplications are required. Thisis generally known as the multiple constant multiplications(MCM) operation and is also a central operation and performancebottleneck in many other DSP systems such as fastFourier transforms, discrete cosine transforms (DCTs), anderror-correcting codes.

Although area-, delay-, and power-efficient multiplier architectures, such as Wallace [2]

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

and modified Booth [3] multipliers, have been proposed, the full flexibility ofmultiplier is not necessary for the constant multiplications, since filter coefficients are fixed and determined beforehand by the DSP algorithms [4]. Hence, the multiplication of filter coefficients with the input data is generally implemented under a shift-adds architecture [5], where each constant multiplication is realized using addition/subtraction and shift operations in an MCM operation [Fig. 1(c)].

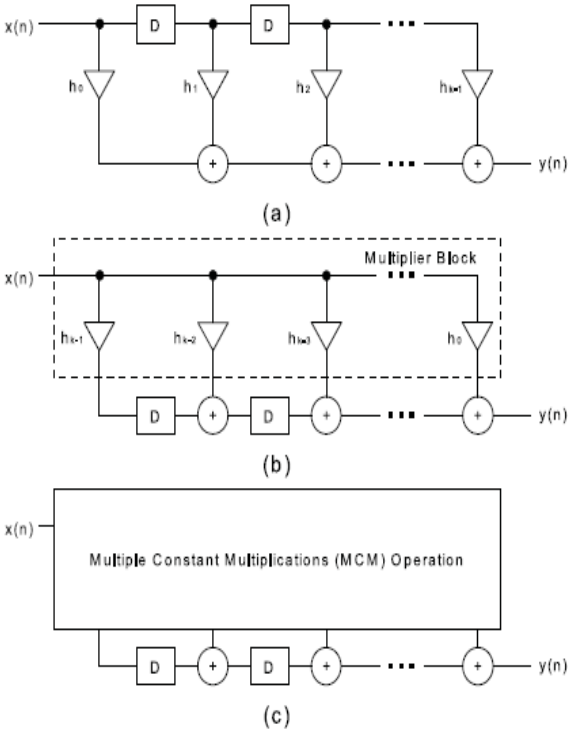

Fig. 1. FIR Filter implementations (a)Direct form. (b)Transposed form. (c)Transposed form with an MCM block

For the shift-adds implementation of constant multiplications,a straightforward method, generally known as digitbasedrecoding [6], initially defines the constants in binary.

Then, for each "1" in the binary representation of the constant,according to its bit position, it shifts the variable and addsup the shifted variables to obtain the result. As a simpleexample, consider the constant multiplications $29x$ and $43x$.Their decompositions in binary are listed as follows:

$29x = (11101)$ binx= $x<<4 + x<<3 + x<<2 + x$

$43x = (101011)$ binx= $x<<5 + x<<3 + x<<1 + x$

which requires six addition operations as illustratedin Fig. 2(a).

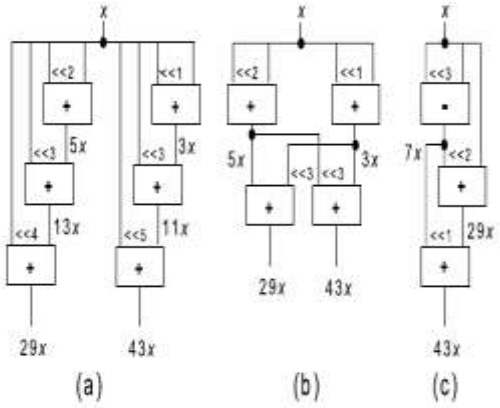

Fig.2.Shift-adds implementations of 29x and 43x (a) Without partial product sharing [6] and with partial product sharing. (b) Without CSE algorithm [9]. (c) Exact GB algorithm

However, the digit-based recoding technique does notexploit the sharing of common partial products, which allowsgreat reductions in the number of operations and, consequently,in area and power dissipation of the MCM design atthe gate level. Hence, the fundamental optimization problem,called the MCM problem, is defined as finding the minimumnumber of addition and subtraction operations that implementthe constant multiplications. Note that, in bit-parallel designof constant multiplications, shifts can be realized using onlywires in hardware without representing any area cost.The

algorithms designed for the MCM problem can becategorized in two classes: common subexpression elimination(CSE) algorithms [7]–[9] and graph-based (GB) techniques[10]–[12]. The CSE algorithms initially extract all possiblesubexpressions from the representations of the constantswhen they are defined under binary, canonical signed digit (CSD) [7], or minimal signed digit (MSD) [8]. Then, they findthe "best" subexpression, generally the most common, to beshared among the constant multiplications. The GB methods are not limited to any particular number representation andconsider a larger number of alternative implementations of aconstant, yielding better solutions than the CSE algorithms, asshown in [11] and [12].

Returning to our example in Fig. 2, the exact CSE algorithmof [9] gives a solution with four operations by finding the mostcommon partial products $3x = (11)_{bin}x$and $5x = (101)_{bin}x$when constants are defined under binary, as illustrated inFig. 2(b). On the other hand, the exact GB algorithm [12]finds a solution with the minimum number of operations bysharing the common partial product $7x$ in both multiplications,as shown in Fig. 2(c). Note that the partial product $7x = (111)_{bin}x$cannot be extracted from the binary representationof $43x$ in the exact CSE algorithm [9].However, all these algorithms assume that the input data$x$ is processed in parallel. On the other hand, in digit-serialarithmetic, the data words are divided into digit sets, consistingof d bits that are processed one at a time [13]. Since digitserialoperators occupy less area and are independent of thedata wordlength, digit-serial architectures offer alternative lowcomplexitydesigns when compared to bit-parallel architectures.However, the shifts require the use of D flip-flops, asopposed to the bit-parallel MCM design where they are freein terms of hardware. Hence, the high-level algorithms shouldtake into account the sharing of shift operations as well as thesharing of addition/subtraction operations in digit-serial MCMdesign. Furthermore, finding the minimum number of operationsrealizing an MCM operation does not always yield anMCM design with optimal area at the gate level [14]. Hence,the high-level algorithms should consider the implementationcost of each digit-serial operation at the gate level.

In this paper, we initially determine the gate-level implementationcosts of digit-serial addition, subtraction, and leftshift operations used in the shift-adds design of digit-serialMCM operations. Then, we introduce the exact CSE algorithm[15] that formalizes the gate-level area optimizationproblem as a 0–1 integer linear programming (ILP) problemwhen constants are defined under a particular numberrepresentation. We also present a new optimization modelthat reduces the 0–1 ILP problem size significantly and,consequently, the runtime of a generic 0–1 ILP solver. Sincethere are still instances which the exact CSE algorithm cannot handle, we describe the approximate GB algorithm [16] thatiteratively finds the "best" partial product which leads to theoptimal area in digit-serial MCM design at the gate level.

In this proposed different graph based multipliers types i.e. CSD – canonic signed-digit multiplier, MSD – minimum signed-digit multiplier MAG – minimum adder graph multiplier,CSDAG – csd adder graph multiplier. Theyare used for low complexity, low power and low area applications.

The section II explains the complexity of serial constant multipliers. Section III explains graph based multipliers. Section IV explains results and analysis.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

II. COMPLEXITY OF SERIAL CONSTANT MULTIPLIERS

In this chapter, the possibilities to minimize the complexity of bit-serialsingle-constant multipliers are investigated [57]. This is done in terms ofthe required number of building blocks, which includes adders and shifts.The multipliers are described using a graph representation. It is shownthat a minimum set of graphs, required to obtain optimal results given certainrestrictions, can be found.In the case of single-constant multipliers, the number of possible solutionscan be limitedbecause of the finite number of graph topologies.However, if a shift-and-add network realizing several coefficients isrequired, a multiple-constant multiplication (MCM) problem is obtained.
Different heuristic algorithms can then be used to reduce the complexity,by utilizing the redundancy between the coefficients. Two algorithms suitableto achieve efficient realization of MCM using serial arithmetic arepresented [56],[62],[66]. It is shown that the new algorithms reduce thetotal complexity significantly.Furthermore, we study the trade-offs in implementations of FIR filtersusing MCM and digit-serial arithmetic. Comparisons considering area,speed, and energy consumption, with respect to the digit-size, are performed[61],[67].

III. Graph Multipliers

In this section, different types of single-constant graph multipliers will bedefined, with respect to constraints on adder cost and throughput. Furthermore,the possibilities to exclude some graphs from the search space areexamined.The investigation covers all coefficients up to 4095 and all types ofgraph multipliers containing up to four adders. All possible graphs, usingthe representation discussed in Section 3.1, for adder costs from 1 to 4are presented in Fig. 2.1 [24].Note that although bit-serial arithmetic will be assumed for the multipliers,results considering adder and flip-flop costs are generally also validfor any digit-serial implementation. However, the numbers of registers thatare required to perform pipelining depend on the digit-size. Furthermore,the cost difference between adders and shifts becomes higher for largerdigit-sizes, since the number of full adders increases linearly while thenumber of flip-flops is constant. Hence, such trade-offs are mainly ofinterest for small digit-sizes.

3.1 Multiplier Types

Different multiplier types can be defined based on the requirements consideringadder cost, flip-flop cost, and pipelining. The types that will bediscussed here are described in the following.

• CSD – Canonic Signed-Digit multiplier
Multiplier based on the CSD representation, as discussed inSection 4.1, with an adder cost equal to one less than the number ofnonzero digits.

• MSD – Minimum Signed-Digit multiplier
Similar to the CSD multiplier and requires the same number of adders,but can in some cases decrease the flip-flop cost by using other MSDrepresentations, which were discussed in Section 3.1.

• MAG – Minimum Adder Graph multiplier
Graph multiplier that is based on any of the topologies in Fig. 4.1 and,for any given coefficient, has the lowest possible adder cost.

Example

To describe the difference between the defined multiplier types, corresponding realizations of the coefficient 2813, which has the CSD representation 1010100000101, are shown in Fig. 4(a). There are other possible solutions for all types except the

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

CSD multiplier. However, note that the values corresponding to the nonzero digits in the CSD representation can be added in different orders, resulting in other structures. Since this may eliminate the pipeline feature, the basic structure used in Fig. 4 (b) will be assumed for CSD multipliers. The adder costs for the multipliers in figs. 4 (a), (b), (c), and are 4, 4, and 3 respectively.

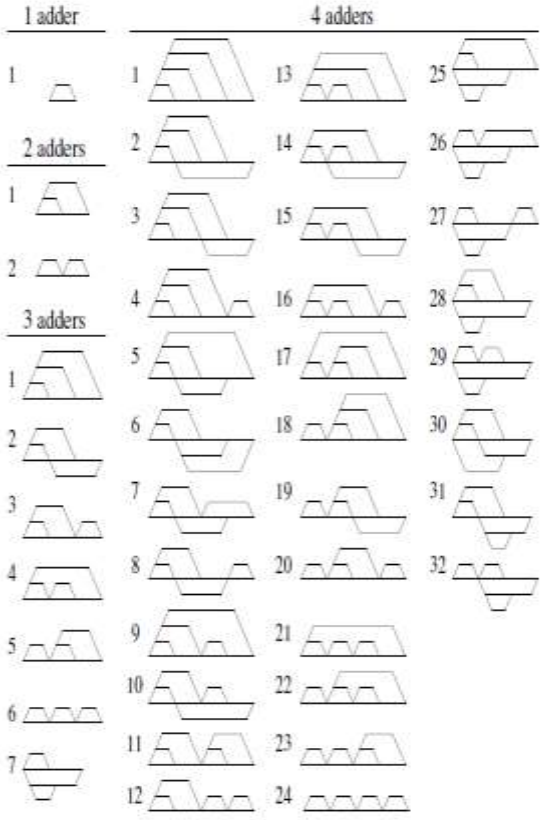

Figure.3. Possible graph topologies for an adder cost up to four.

This implies that it is possible to save either two shifts, or one adder and oneShift compared to the CSD multiplier.4 (b) and (c) with an extra cost of 0 and 1 register, respectively. Note that the flip-flop cost will include both shifts and pipelining registers,since both correspond to a single flip-flop in bit-serial arithmetic.

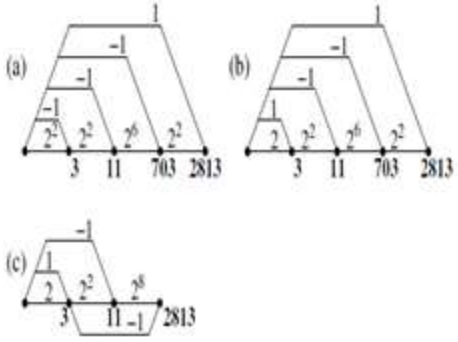

Figure.4. Differentrealizations of the coefficient 2813. (a) CSD, (b) MSD, (c) MA G

IV.RESULTS AND ANALYSIS

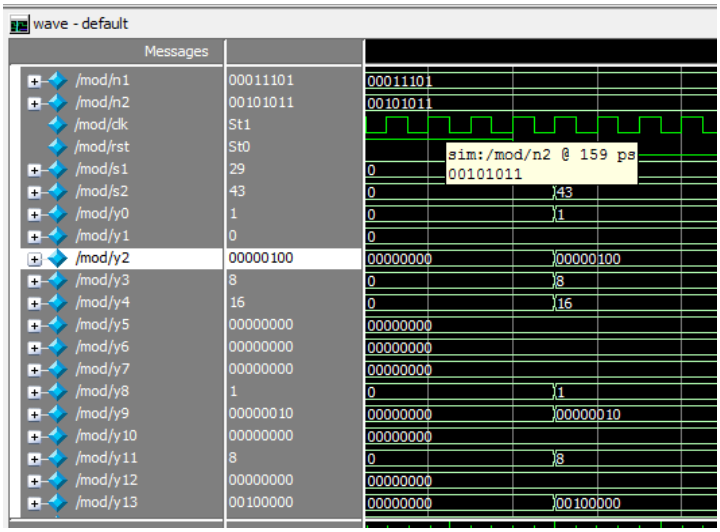

Fig.5.Waveform Results of GB Algorithm.

TABLE 1

The comparison Result of CSE and BE Algorithm

Algorithm name	Delay(ns)	Area(%)
CSE	2.780	33
GB	2.58	30

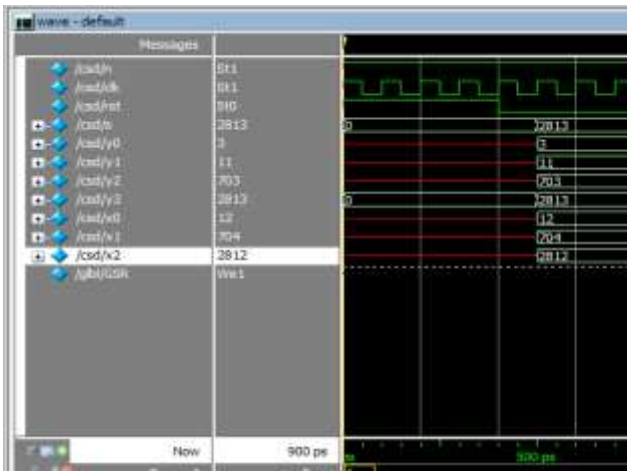

Fig.6. Waveform Results of CSDAG Algorithm

TABLE 2

The Comparison Result of CSD and MAG

Graph based Multipliers	Delay(ns)	Area (%)
CSD	2.58	24
MAG	2.58	19

The comparison tables and waveforms show the analysis of different algorithms and different graph based multipliers.in this BE algorithm shows an efficient results compare to CSE algorithms. Again analyses in graph based multipliers i.e. CSD, MSD, and MAG.the MAG shows good area in table 2.

V.CONCLUSION

The proposed new approach is MAG for implementing reconfigurable higher order filters with low complexity. The proposed MAGmethod make use of architecture with fixed number of multiplexers and the reduction in complexity is achieved by applying the graph based algorithm. The MAG architecture results in high speed filters and low area and thus low power filter implementations. The MAG also provides the flexibility of changing the filter coefficient word lengths dynamically. The proposed reconfigurable architectures can be easily modified to employ any graph based (GB) method, which results in architectures that offers good area and power reductions and speed improvement reconfigurable FIR filter implementations.

REFERENCES

[1] L. Wanhammar, *DSP Integrated Circuits*. New York: Academic, 1999.

[2] C. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron.Comput.*, vol. 13, no. 1, pp. 14–17, Feb. 1964.

[3] W. Gallagher and E. Swartz lander, "High radix booth multipliers usingreduced area adder trees," in *Proc. Asilomar Conf. Signals, Syst. Comput.*,vol. 1. Pacific Grove, CA, Oct.–Nov. 1994, pp. 545–549.

[4] J. McClellan, T. Parks, and L. Rabiner, "A computer program fordesigning optimum FIR linear phase digital filters," *IEEE Trans. AudioElectroacoust.*, vol. 21, no. 6, pp. 506–526, Dec. 1973.

[5] H. Nguyen and A. Chatterjee, "Number-splitting with shift-and-adddecomposition for power and hardware optimization in linear DSPsynthesis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8,no. 4, pp. 419–424, Aug. 2000.

[6] M. Ercegovac and T. Lang, *Digital Arithmetic*. San Mateo, CA: MorganKaufmann, 2003.

[7] R. Hartley, "Subexpression sharing in filters using canonic signed digitmultipliers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 43, no. 10,pp. 677–688, Oct. 1996.

[8] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on minimalsigned digit representation," in *Proc. DAC*, 2001, pp. 468–473.

[9] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Exact and approximatealgorithms for the optimization of area and delay in multiple constantmultiplications," *IEEE Trans. Comput.-Aided Design Integr.CircuitsSyst.*, vol. 27, no. 6, pp. 1013–1026, Jun. 2008.

[10] A. Dempster and M. Macleod, "Use of minimum-adder multiplier blocksin FIR digital filters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 42,no. 9, pp. 569–577, Sep. 1995.

[11] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication,"*ACM Trans. Algor.*, vol. 3, no. 2, pp. 1–39, May 2007.

[12] L. Aksoy, E. Gunes, and P. Flores, "Search algorithms for the multipleconstant multiplications problem: Exact and approximate," *J. Microprocess.Microsyst.*, vol. 34, no. 5, pp. 151–162, Aug. 2010.

[13] R. Hartley and K. Parhi, *Digit-Serial Computation*. Norwell, MA:Kluwer, 1995.

[14] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Optimization of areain digital FIR filters using gate-level metrics," in *Proc. DAC*, 2007, pp.420–423.

[15] L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro, "Optimization of area in digit-serial multiple constant multiplications at gate-level," in*Proc. ISCAS*, 2011, pp. 2737–2740.

[16] L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro, "Efficientshift-adds design of digit-serial multiple constant multiplications," in*Proc. Great Lakes Symp. VLSI*, 2011, pp. 61–66.

[17] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. New York: Oxford Univ. Press, 2000.

[18] K. Johansson, O. Gustafsson, A. G. Dempster, and L. Wanhammar,"Algorithm to reduce the number of shifts and additions in multiplierblocks using serial arithmetic," in *Proc. IEEE MediterraneanElectrotechnicalConf.*, Dubrovnik, Croatia, May 12–15, 2004, vol. 1, pp. 197– 200.

[19] K. Johansson, O. Gustafsson, and L. Wanhammar, "Implementation oflow-complexity FIR filters using serial arithmetic," in *Proc. IEEE Int.Symp. Circuits Syst.*, Kobe, Japan, May 23–26, 2005, vol. 2, pp. 1449–1452.

[20] K. Johansson, O. Gustafsson, A. G. Dempster, and L. Wanhammar, "Trade-offs in low power multiplier blocks using serial arithmetic," in *Proc. National Conf. Radio Science (RVK)*, Linköping, Sweden, June 14–16, 2005, pp. 271–274.

[21] K. Johansson, O. Gustafsson, and L. Wanhammar, "Trade-offs in multiplier block algorithms for low power digit-serial FIR filters," in *Proc.WSEAS Int. Conf.*

Circuits, Vouliagmeni, Greece, July 10–12, 2006

[22] K. Johansson, O. Gustafsson, and L. Wanhammar, "Multiple constantmultiplication for digit-serial implementation of low power FIR filters," *WSEAS Trans. Circuits Syst.*, vol. 5, no. 7, pp. 1001–1008, July 2006..

[24] A. G. Dempster and M. D. Macleod, "Constant integer multiplicationusing minimum adders," *IEE Proc. Circuits Devices Syst.*, vol. 141.